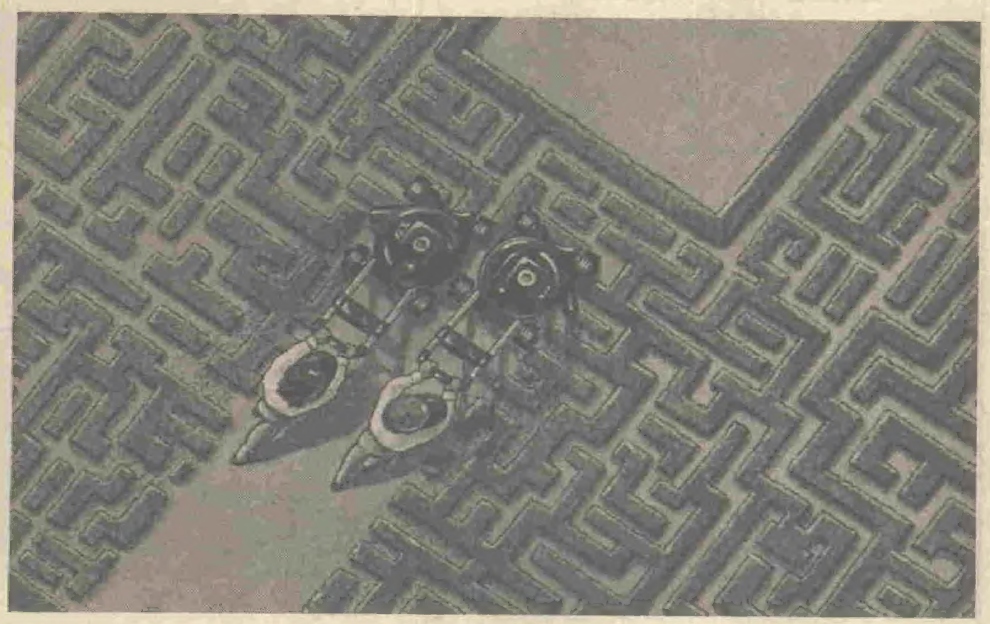
สิทธิพงศ์ จันทร์แสงอรุณ

ไคล์เอ็นต์เซอร์ฟเวอร์ในระบบคอมพิวเตอร์



ศึกษารูปแบบการคำนวณแบบกระจายศูนย์ที่มีประสิทธิภาพในการทำงานแตกต่างกันไป

ในหน่วยงานและสถานศึกษาในปัจจุบัน คงไม่มีใครปฏิเสธได้ว่าระบบเครือข่ายได้เข้ามามีบทบาทต่อชีวิตประจำวันของพนักงานและนักศึกษาเป็นอย่างมาก ระบบเครือข่ายได้เริ่มเข้ามาแทนที่ระบบใหญ่ ๆ อย่างเมนเฟรมและมินิคอมพิวเตอร์แล้ว สาเหตุที่มันสามารถเข้ามาแทนที่ได้ก็คือ การคำนวณแบบกระจายศูนย์หรือที่เรียกว่า distributed processing นั่นเอง ซึ่งสามารถแบ่งงานกันทำในแต่ละเวอร์กสเตชันได้ ในขณะที่เครื่องเมนเฟรมและมินิคอมพิวเตอร์ส่วนใหญ่ยังใช้การคำนวณแบบ central processing หรือระบบการคำนวณจากส่วนกลางอยู่

ปัจจุบันผู้ประกอบการในเครื่องระดับเมนเฟรมและมินิคอมพิวเตอร์เริ่มหันมาวางโครงสร้างของตัวเองใหม่ให้มาเป็นรูปแบบการคำนวณแบบกระจายศูนย์กันแล้ว ซึ่งมีหลายแบบ แต่ละแบบมีประสิทธิภาพที่แตกต่างกัน แต่แบบที่กำลังนิยมกันอยู่ในขณะนี้ก็คือแบบที่เรียกว่า Client/Server processing

คุณอาจจะได้ยินคำกล่าวที่ว่าการทำงานแบบไคล์เอ็นต์เซอร์ฟเวอร์ก็คือหนทางในการคำนวณในระบบคอมพิวเตอร์ในทศวรรษที่ 90 คำกล่าวที่ว่านี้มีส่วนเป็นจริงมากทีเดียว เพราะระบบที่เป็นไคล์เอ็นต์เซอร์ฟเวอร์จะช่วยลดค่าใช้จ่ายของการบำรุงรักษาซอฟต์แวร์ได้มาก (maintenance costs) สามารถช่วยให้คุณไม่ติดอยู่กับซอฟต์แวร์หรือแอพพลิเคชันเฉพาะกิจบางประเภท ช่วยเพิ่มประสิทธิภาพการทำงานของระบบเครือข่าย และสามารถช่วยให้คุณพ้นไปจากระบบเมนเฟรมและมินิคอมพิวเตอร์อันแสนจะยุ่งยากและแพง

ก่อนที่เราจะมาพิจารณาถึงความสามารถและการทำงานของระบบไคล์เอ็นต์เซอร์ฟเวอร์ ผู้เขียนขออธิบายถึงการทำงานในแบบอื่น ๆ ก่อนที่ระบบต่าง ๆ จะถูกพัฒนาขึ้นมาให้อยู่ในระบบไคล์เอ็นต์เซอร์ฟเวอร์

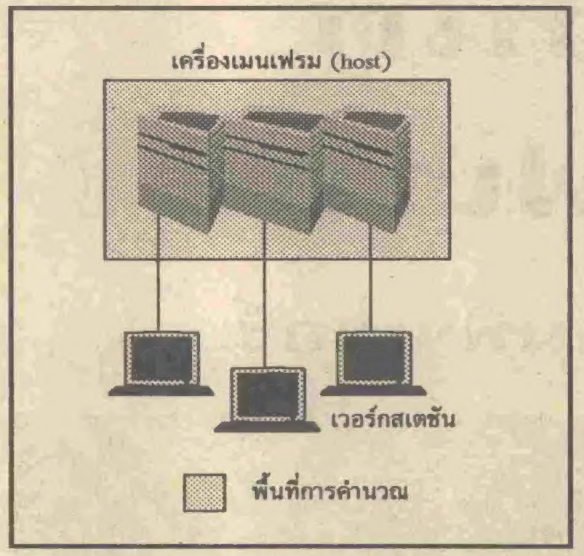
Host-Based processing

ระบบแบบนี้เป็นการทำงานในระบบดั้งเดิมของเมนเฟรมที่การคำนวณทุกอย่างจะต้องกลับเข้ามาที่ตัวโฮสต์

ทั้งหมด ซึ่งมันจะทำการคำนวณและส่งผลผ่านหน้าจอกลับไปให้ผู้ใช้ รูปที่ 1 แสดงให้เห็นถึงการทำงานของระบบนี้

จากรูปที่ 1 คุณจะเห็นได้ว่าระบบแบบนี้มีการต่อเชื่อมกับตัวโฮสต์โดยตรง ซึ่งเวอร์กสเตชันส่วนใหญ่จะเป็นประเภทดั้มเทอร์มินัลที่ไม่สามารถคำนวณอะไรได้เลย ทำได้เพียงแต่จัดการกับระบบการแสดงผลและการส่งรับข้อมูลจากโฮสต์เท่านั้น

แอพพลิเคชันทั้งหมดจะวิ่งอยู่ในโฮสต์ เมื่อผู้ใช้ต้องการข้อมูลบางอย่าง โฮสต์จะเป็นตัวจัดการหาให้และส่งเพียงข้อมูลจากจอภาพกลับไปแจ้งแก่ผู้ใช้เท่านั้น จะสังเกตได้ว่าสัญญาณต่าง ๆ ที่ส่งภาพไปมาในระบบเป็นเพียงชุดข้อมูลของหน้าจอภาพเท่านั้น ระบบแบบนี้จึงไม่มีความจำเป็นที่จะต้องมีความเร็วของระบบเครือข่ายที่สูงมากนัก ตัวอย่างการทำงานในเครื่องเมนเฟรมของ IBM คือ โปรแกรม Virtual Telecommunication Access Method (VTAM) ที่อยู่ในเครื่องเมนเฟรม และโปรแกรม Network Control Project (NCP) อยู่ในตัว 3745 ซึ่งเป็นอุปกรณ์ทางด้านการสื่อสารของ IBM

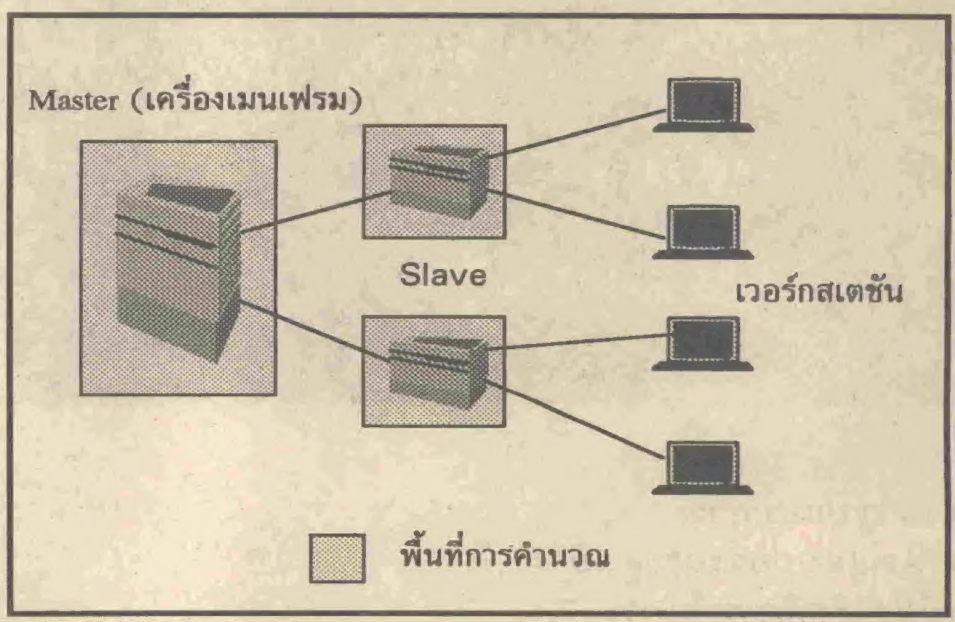


รูปที่ 1
การทำงานของระบบ Host-Based Processing

Master-Slave processing

เป็นระบบที่มีการส่งผ่านข้อมูลแบบ Master/Slave การทำงานส่วนใหญ่ของ Slave จะต้องถูกควบคุมโดย Master ของมัน รูปที่ 2 แสดงลักษณะการทำงานของ Master-Slave processing

จะเห็นได้ว่าการทำงานของมันจะคล้าย ๆ กับการทำงานแบบ Hostbased processing และมันจะมีการกระจายการคำนวณออกบ้าง แต่ยังไม่ถึงในระดับเวอร์กสเตชัน ตัวโฮสต์ซึ่งทำหน้าที่เป็น Master จะส่งผ่านข้อมูลบางอย่างไปที่ Slave ของมัน เพื่อทำงานกับเวอร์กสเตชันได้รวดเร็วขึ้น



รูปที่ 2 การทำงานแบบ Master/Slave

จากรูปที่ 2 แอพพลิเคชันทั้งหมดก็ยังคงทำงานอยู่ในเครื่องเมนเฟรมหรือ Master ซึ่งจะมีเวลาในการคำนวณมากขึ้น เพราะมันเพียงแต่ต้องควบคุม Slave ของมันเพียงไม่กี่ตัวเท่านั้น ส่วนเวอร์กสเตชันจะถูกควบคุมโดยซีพียูของ Slave อีกต่อหนึ่ง เวอร์กสเตชันยังคงทำหน้าที่เพียงแต่แสดงผลทางหน้าจอเท่านั้น ยังไม่มีการคำนวณใด ๆ เพื่อช่วยเหลือโฮสต์เลย

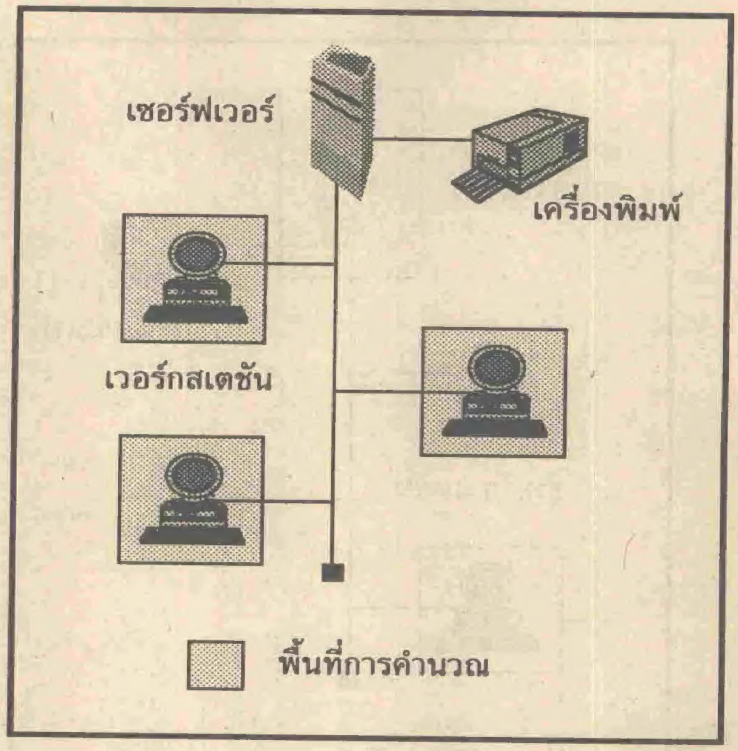
วิธีนี้จะช่วยทำให้ระบบการต่อเชื่อมเป็นไปได้ง่ายขึ้น โดยการแบ่งกลุ่มของผู้ใช้เป็นจุด ๆ แทนการต่อเชื่อมออกมาจากจุดเดียวกันของ Host-based processing ตัวอย่างคือ เครื่อง IBM 30XX

ที่ใช้กับ cluster controller และ intelligent terminal

Client/Server processing

ส่วนใหญ่คุณจะพบการทำงานของระบบไคล์เอ็นต์เซอร์ฟเวอร์นี้ในระบบเครือข่ายแลน ระบบนี้ยังแบ่งออกได้เป็นสองชนิดคือ Shared-device processing และแบบ Client/Server processing

ระบบแบบ Shared-device processing ถูกพัฒนาขึ้นมาก่อนเพื่อตอบสนองการขยายตัวอย่างรวดเร็วของระบบเครือข่ายแลน ระบบแบบนี้มีข้อดีและมีความรวดเร็วมากกว่าระบบอื่น ๆ ที่เราได้กล่าวมาแล้วข้างต้น การทำงานของระบบนี้แสดงอยู่ในรูปที่ 3



รูปที่ 3 การทำงานแบบ Shared-device

ความแตกต่างที่เห็นได้ชัดก็คือแอพพลิเคชันแทนที่จะวิ่งอยู่บนโฮสต์ ซึ่งในระบบแบบนี้จะเรียกโฮสต์ว่าเซอร์ฟเวอร์ การทำงานส่วนใหญ่ของมันจะยอมให้ผู้ใช้มาร่วมใช้รีซอร์สต่าง ๆ ที่อยู่บนเซอร์ฟเวอร์ได้ แต่รีซอร์สจะถูกจำกัดให้เป็นเพียงการร่วมกันใช้ไฟล์และเครื่องพิมพ์เท่านั้น

ข้อเสียของมันก็คือ การทำงานส่วนใหญ่จะอยู่ในเครื่องเวอร์กสเตชันของผู้ใช้เท่านั้น ยกเว้นฟังก์ชันทางการพิมพ์และการทำไฟล์ I/O ที่จะอยู่บนเครื่องเซอร์ฟเวอร์ เมื่อผู้ใช้ต้องการใช้แอพพลิเคชันหรือไฟล์ มันจะทำการร้องขอมาที่เซอร์ฟเวอร์ เซอร์ฟเวอร์จะส่งไฟล์ทั้งหมดหรือแอพพลิเคชันทั้งตัวผ่านระบบเครือข่ายไปให้เวอร์กสเตชันของผู้ใช้ จากนั้นเวอร์กสเตชันก็จะโหลดเอาไฟล์หรือแอพพลิเคชันนั้น ๆ เข้ามาทำงานต่อไป เซอร์ฟเวอร์ที่ทำหน้าที่ในการแบ่งไฟล์กันใช้นี้เรียกว่า File server

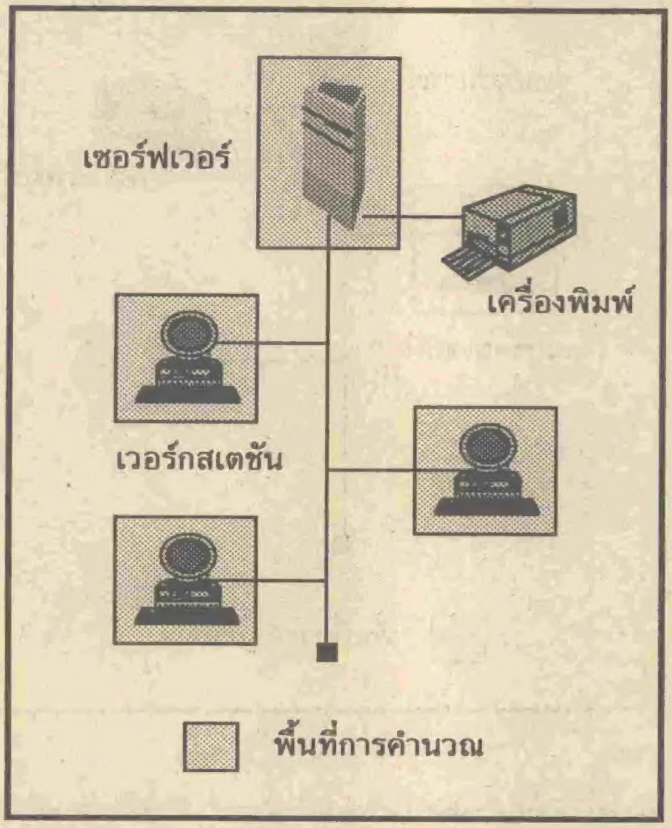
ฟังก์ชันทางการพิมพ์ก็คล้าย ๆ กันคือ เมื่อผู้ใช้ต้องการพิมพ์ไฟล์ข้อมูล มันก็จะส่งไฟล์นั้นไปให้เซอร์ฟเวอร์เพื่อส่งออกมาพิมพ์ เซอร์ฟเวอร์ที่ทำหน้าที่นี้จะเรียกว่า Print server บางครั้งเซอร์ฟเวอร์ตัวเดียวสามารถทำหน้าที่ของ File server และ Print server ได้

ระบบเครือข่ายที่สามารถรองรับการทำงานแบบนี้ได้จะต้องเป็นระบบที่มีความเร็วพอสมควร เพราะข้อมูลที่วิ่งไปมาในระบบมีขนาดใหญ่ (ส่งรับข้อมูลทั้งไฟล์หรือดาต้าเบส) ระบบที่นิยมใช้โดยทั่วไปคือ Novell Netware และ Microsoft LAN Manager เป็นต้น ทั้งสองตัวจะวิ่งโดยมาตรฐานหลายอย่างรวมทั้ง Ethernet ที่มีความเร็ว 10 MBps หรือ Token Ring ที่มีความเร็ว 4 หรือ 16 MBps

เซอร์ฟเวอร์ที่ใช้ในระบบไม่จำเป็นต้องมีขนาดซีพียูที่ใหญ่โตซึ่งมีราคาแพง อย่างในระบบเมนเฟรมและมินิคอมพิวเตอร์อีกต่อไปแล้ว แต่เครื่องเวอร์กสเตชันจำเป็นจะต้องมีซีพียูที่สามารถวิ่งแอพพลิเคชันของตนเองได้ ความเร็วของการทำงานก็จะถูกจำกัดอยู่ในเครื่องเวอร์กสเตชันนั่นเอง

ระบบ Client/Server processing จะคล้ายกับระบบ Shared-device processing มาก แต่แทนที่การคำนวณทางด้านแอพพลิเคชันจะอยู่ในเครื่องเวอร์กสเตชันของผู้ใช้ทั้งหมด มันจะแบ่งการคำนวณมาอยู่ในตัวเซอร์ฟเวอร์ด้วย ดังที่แสดงไว้ในรูปที่ 4

จะเห็นว่าแอพพลิเคชันทำงานทั้งในเครื่องเซอร์ฟเวอร์และเครื่องเวอร์กสเตชันด้วย เมื่อผู้ใช้ต้องการข้อมูลบางส่วนที่อยู่ในดาต้าเบสไฟล์ขนาดใหญ่ เครื่องเวอร์กสเตชันจะส่งคำร้องขอข้อมูลนั้นไปที่ตัวเซอร์ฟเวอร์ เซอร์ฟเวอร์จะทำการคำนวณและดึงข้อมูลเหล่านั้นออกมา และส่งเฉพาะข้อมูลนั้น ๆ กลับมายังเครื่องเวอร์กสเตชันที่ร้องขอมา เมื่อเวอร์กสเตชันได้รับข้อมูลมันก็จะนำมาคำนวณต่อไป



รูปที่ 4
การทำงานแบบ Client/Server

จะเห็นได้ว่าข้อมูลที่ส่งไปมาในระบบจะมีเพียงข้อมูลที่จำเป็นเท่านั้น ในขณะที่ถ้าส่งในระบบ Shared-device processing จะเป็นการส่งไฟล์ดาต้าเบสใหญ่ ๆ ทั้งไฟล์มาที่เครื่องเวอร์กสเตชันทั้ง ๆ ที่มันต้องการข้อมูลเพียงไม่กี่ไบต์ก็ตาม เมื่อสถาบันหรือหน่วยงานใดมีการใช้แอพพลิเคชันแบบ Shared-device บนระบบเครือข่ายมากเกินไป จะทำให้ระบบเครือข่ายนั้น ๆ ไม่สามารถรองรับ traffic ที่มากเช่นนั้นได้ เวลาในการตอบสนองของระบบต่อผู้ใช้ก็จะมีค่าสูงขึ้นด้วย แต่ถ้าหน่วยงานใดหันมาใช้แอพพลิเคชันแบบไคล์เอ็นต์เซอร์ฟเวอร์มากขึ้น คุณก็ไม่จำเป็นต้องมากังวลถึง traffic ที่จะเกิดขึ้นในระบบเครือข่ายของคุณเลย เพราะระบบไคล์เอ็นต์-เซอร์ฟเวอร์จะใช้ traffic น้อยมากเมื่อเทียบกับระบบแบบ Shared-device

ด้วยการส่งข้อมูลที่มีประสิทธิภาพแบบนี้ ทำให้การติดต่อกันข้าม WAN นี้สามารถทำได้รวดเร็วมาก ความสามารถและประสิทธิภาพในระบบก็มีสูงขึ้นด้วย อีกทั้งแอพพลิเคชันที่วิ่งอยู่บนตัวเซอร์ฟเวอร์ที่คอยให้บริการทางด้านข้อมูลแก่ผู้ใช้หรือไคล์เอ็นต์ก็ไม่จำเป็นจะต้องเป็นแอพพลิเคชันแบบเดียวกัน แต่ขอให้มีมาตรฐานแบบเดียวกันก็พอ ตัวอย่างของแอพพลิเคชันที่วิ่งอยู่ในเซอร์ฟเวอร์ก็คือ ดาต้าเบสซอฟต์แวร์อย่าง Sybase และ Microsoft SQL Server เป็นต้น แอพพลิเคชันทางดาต้าเบสเหล่านี้จะอยู่บน NOS (Network Operating System) อย่าง Netware และ MS LAN Manager ที่เวอร์กสเตชันของไคล์เอ็นต์ผู้ใช้สามารถใช้แอพพลิเคชันอย่าง MS Access, Excel, และ Paradox เป็นต้น เพื่อทำหน้าที่ร้องขอบริการจากเซอร์ฟเวอร์

ระบบไคล์เอ็นต์เซอร์ฟเวอร์ยังสามารถแบ่งอย่างละเอียดได้เป็น 5 ชนิด คือ

1. ชนิดการกระจายการแสดงผล (Distributed Presentation)

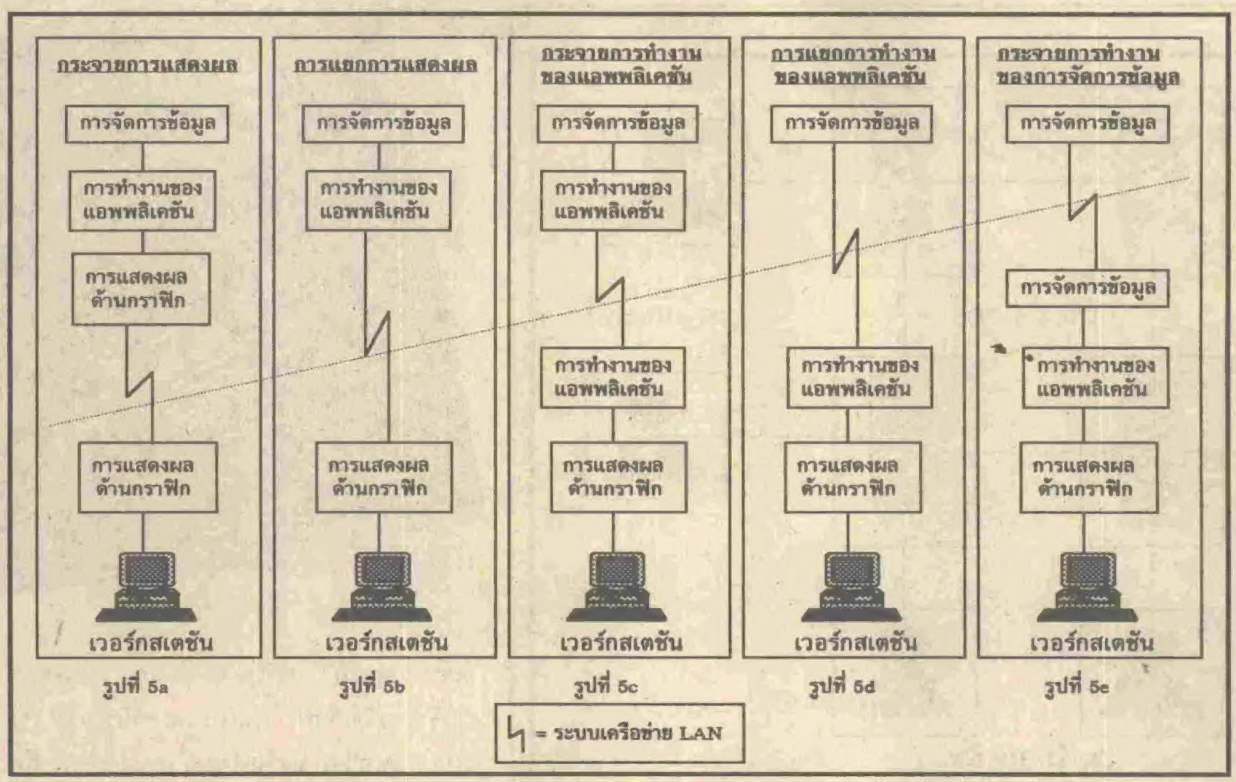
เป็นการทำงานภายใต้ Graphical User Interface หรือ GUI อย่างในวินโดวส์ เซอร์ฟเวอร์จะทำการคำนวณและส่งภาพกราฟิกที่ใช้ติดต่อกับผู้ใช้มาที่เครื่องเวอร์กสเตชันเมื่อมันร้องขอมา ในขณะเดียวกันผู้ใช้ก็จะไม่สามารถบอกความแตกต่างได้เลยว่าภาพที่ปรากฏอยู่นั้นเป็นภาพที่มาจากเซอร์ฟเวอร์หรือจากในตัวเครื่องเอง การทำงานของระบบไคล์เอ็นต์เซอร์ฟเวอร์ชนิดนี้จะเป็นแบบที่กระจายการทำงานน้อยที่สุดของทั้งห้าชนิด (รูปที่ 5a)

2. ชนิดแยกการแสดงผล (Remote Presentation)

จากรูปที่ 5b จะเห็นว่าการแสดงผลต่าง ๆ ต่อผู้ใช้จะถูกทำจากเวอร์กสเตชันหรือไคล์เอ็นต์เท่านั้น การทำงานของแอพพลิเคชันลอจิกจะอยู่ในเซอร์ฟเวอร์ทั้งหมด รวมทั้งการจัดการข้อมูลด้วย ข้อมูลที่ส่งข้ามระบบเครือข่ายก็จะมีน้อยกว่าชนิดที่หนึ่งเพราะว่ามันไม่ต้องส่งภาพกราฟิกที่มีขนาดใหญ่ข้ามระบบไปให้ผู้ใช้เลย

3. ชนิดการกระจายการทำงานของแอพพลิเคชัน (Distributed Logic)

ระบบชนิดนี้จะมีการกระจายการทำงานของตัวแอพพลิเคชันเองให้อยู่ในทั้งตัวเซอร์ฟเวอร์และเวอร์กสเตชันดังเช่นในรูปที่ 5c จะเห็นได้ว่าเวอร์กสเตชันจะ



รูปที่ 5
ชนิดต่าง ๆ ของระบบแบบไคล์เอ็นต์เซอร์ฟเวอร์

ช่วยเซอร์ฟเวอร์ทำงานทางด้านแอพพลิเคชันเป็นส่วนหน้า (Front-End) ซึ่งมีการเชื่อมโยงกับแอพพลิเคชันส่วนหลัง (Back-End) ที่อยู่ในตัวเซอร์ฟเวอร์ด้วย การคำนวณทางด้านแอพพลิเคชันก็จะถูกแบ่งออกจากเซอร์ฟเวอร์มากขึ้น ตัวอย่างเช่น ในเวอร์กสเตชันมีการคำนวณหลายชุด มันก็อาจส่งการคำนวณบางส่วนหรือทั้งหมดไปให้เซอร์ฟเวอร์คำนวณและจะให้ส่งผลลัพธ์กลับมาให้มัน

4. ชนิดแยกการทำงานของแอพพลิเคชัน (Remote Date Management)

การทำงานของแอพพลิเคชันทั้งหมดจะอยู่ในตัวเวอร์กสเตชันทั้งหมด และเซอร์ฟเวอร์จะมีส่วนการจัดการข้อมูลเท่านั้น เมื่อเวอร์กสเตชันต้องการข้อมูลส่วนใดก็สามารถทำการร้องขอไปที่เซอร์ฟเวอร์ได้ เซอร์ฟเวอร์จะทำการค้นหาข้อมูลให้และส่งกลับมาที่เครื่องเวอร์กสเตชันดังรูปที่ 5d

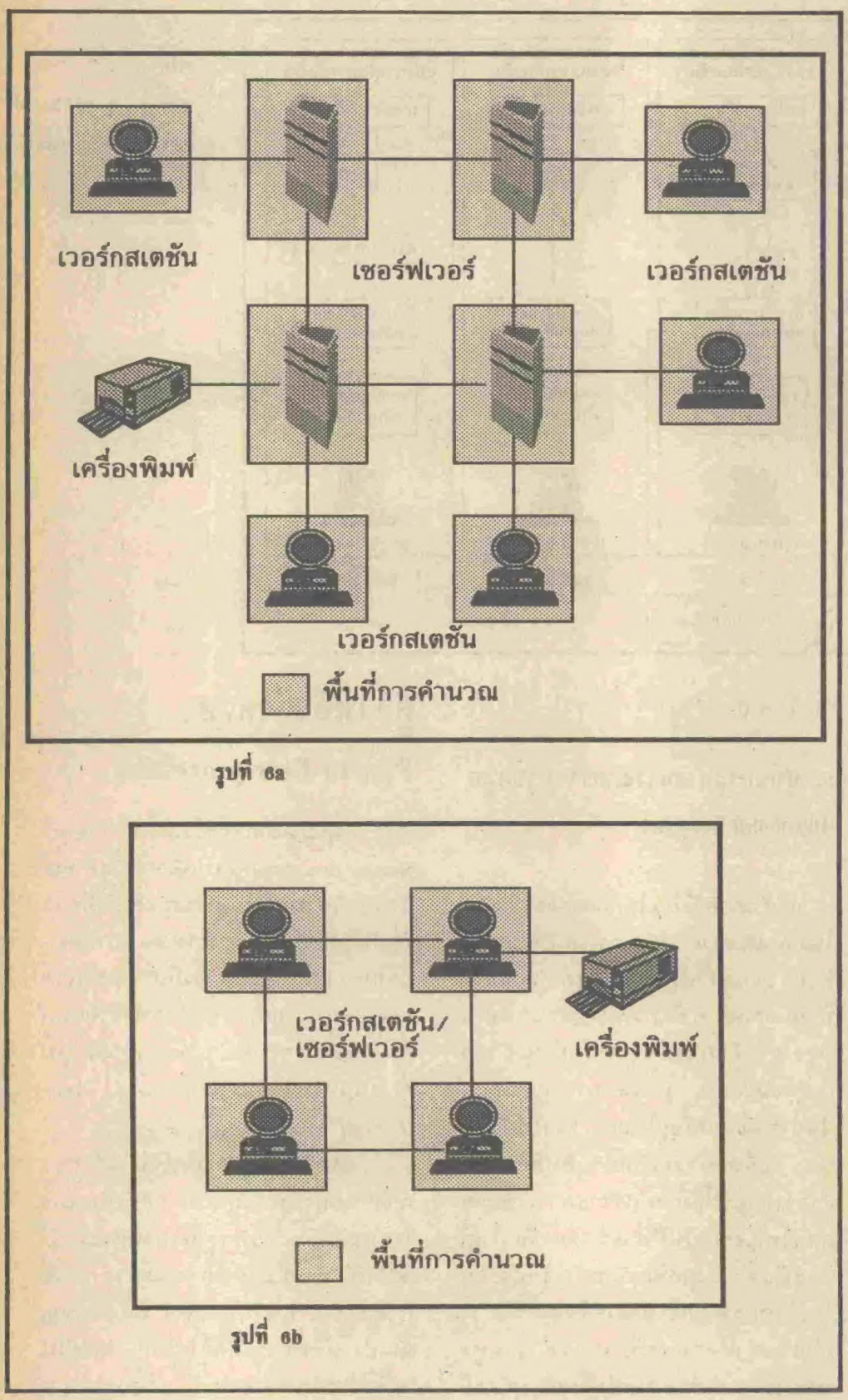
5. ชนิดกระจายการจัดการของข้อมูล (Distributed Database)

เป็นชนิดที่นิยมใช้กันอย่างแพร่หลายในแอพพลิเคชัน Sybase หรือ Microsoft SQL Server เป็นต้น การคำนวณทางด้านแอพพลิเคชันทั้งหมดและบางส่วนของการจัดการข้อมูลจะอยู่ในเครื่องเวอร์กสเตชัน และส่วนหลังของการจัดการข้อมูลจะอยู่ในเซอร์ฟเวอร์ดังรูป 5e วิธีนี้สามารถเพิ่มประสิทธิภาพตัวเซอร์ฟเวอร์ขึ้นมาก เพราะการคำนวณส่วนใหญ่จะอยู่ในตัวเวอร์กสเตชัน แต่การค้นหาข้อมูลที่มีความซับซ้อนจะอยู่ในตัวเซอร์ฟเวอร์ แต่เวอร์กสเตชันอาจต้องมีความสามารถในการทำงานสูงพอควร การทำงานของทั้งระบบก็จะมีความรวดเร็วมาก

สู่สิ่งที่ดีกว่าด้วย Peer-to-Peer processing

ถ้าคุณคิดว่าการทำงานแบบ Client/Server processing เป็นสิ่งที่ดีที่สุด คุณก็คงคิดผิดไปถนัด เพราะขณะนี้มีการตื่นตัวกันมากกับการทำงานแบบ Peer-to-Peer processing ซึ่งมีประสิทธิภาพของการทำงานมากกว่าการทำงานแบบ Client/Server processing เสียอีก รูปที่ 6 แสดงการทำงานของระบบ Peer-to-Peer processing

จากรูปที่ 6 คุณจะเห็นได้ว่าการทำงานของมันจะถูกแบ่งออกเป็นสองแบบ แต่หลักการทำงานของมันจะใกล้เคียงกันมากและมีพื้นฐานมาจากการทำงานของ Client/Server processing นั่นเอง แทนที่จะมีแต่ผู้ใช้ที่ทำงานอยู่ในเครื่องเวอร์กสเตชันที่จะร้องขอการบริการจากเซอร์ฟเวอร์เท่านั้น ตัว



รูปที่ 6a

รูปที่ 6b

รูปที่ 6 การทำงานแบบ Peer-to-Peer processing

เซอร์ฟเวอร์เองก็สามารถร้องขอการบริการได้ด้วย ในแบบแรกที่เห็นในรูป 6a เซอร์ฟเวอร์แต่ละตัวจะต่อเชื่อมถึงกันและสามารถร้องขอข้อมูลจากเซอร์ฟเวอร์ตัวอื่น ๆ ได้ ผู้ใช้ไม่จำเป็นต้องรู้เลยว่าข้อมูลที่ตนเองต้องการอยู่ในเซอร์ฟเวอร์ตัวใด เมื่อมันร้องขอข้อมูลไป เซอร์ฟเวอร์แต่ละตัวจะทำหน้าที่เป็นไคล์เอ็นต์และเซอร์ฟเวอร์สลับกันไปมาเพื่อค้นหาข้อมูลที่ต้องการ เมื่อได้มาแล้วมันก็จะทำตัวเป็นเซอร์ฟเวอร์อีกครั้งเพื่อส่งข้อมูลที่ได้มากลับไปให้เวอร์กสเตชัน-ไคล์เอ็นต์ที่ร้องขอมา

การทำงานแบบที่สอง (รูปที่ 6b) เวอร์กสเตชันแต่ละตัวจะทำหน้าที่ได้ทั้งเซอร์ฟเวอร์และไคล์เอ็นต์ ซึ่งก็หมายความว่าแต่ละตัวสามารถทำการร้องขอและให้บริการตัวอื่น ๆ ได้ในเวลาเดียวกัน จะสังเกตว่าในแบบนี้เวอร์กสเตชันทุกตัวจะอยู่ในระดับเดียวกันและไม่มีการแยกกลุ่มกันเหมือนกับในแบบอื่น ๆ ดังนั้นในระบบนี้จะเป็นการร้องขอการบริการกันโดยตรงทั้งสองทางแทนที่จะต้องผ่านการร้องขอไปให้เครื่องเวอร์กสเตชันตัวอื่น ๆ

ตารางที่ 1 เป็นการเปรียบเทียบความสามารถของระบบแบบไคล์เอ็นต์เซอร์ฟเวอร์กับระบบแบบอื่น ๆ

Client/Server Processing	การทำงานในแบบอื่น
• มีประสิทธิภาพการทำงานของระบบสูง สามารถส่งข้อมูลได้มากและรวดเร็วกว่าเพราะส่งข้อมูลที่จำเป็นไปตามที่ client ร้องขอเท่านั้น	• ทำงานได้ช้าเพราะการคำนวณจะรวมอยู่ที่เดียวคือที่ host หรือมีการส่งไฟล์ทั้งหมดไปมาในระบบเครือข่าย
• ควบคุมการทำงานโดย client หรือผู้ใช้บริการ	• ควบคุมการทำงานจาก host
• มีการแบ่งการทำงานระหว่างเซอร์ฟเวอร์และผู้ใช้	• ทำการคำนวณทางด้านแอพพลิเคชันอยู่แต่ใน host
• สามารถเลือกใช้แอพพลิเคชันที่แตกต่างกับแอพพลิเคชันที่อยู่ในตัวเซอร์ฟเวอร์ได้	• ใช้แอพพลิเคชันได้อย่างจำกัด
• ระบบสามารถขยายไปได้อีกเรื่อย ๆ เพราะมีการกระจายกันทำงานในแต่ละเวอร์กสเตชัน	• เมื่อทำการขยายระบบอาจจะต้องทำการเปลี่ยนแปลงไปใช้ CPU ที่มีประสิทธิภาพสูงขึ้นหรือระบบเครือข่ายที่มีความเร็วมากขึ้นซึ่งจะมีราคาในการขยายระบบแพงมากกว่า

บทส่งท้าย

ระบบใหญ่ ๆ อย่างเมนเฟรมและมินิคอมพิวเตอร์เริ่มหันมาให้ความสนใจและพัฒนาระบบ Client/Server และ Peer-to-Peer processing กันมากขึ้น เช่น ระบบ AS/400 ของ IBM ที่นิยมใช้กันมากในบ้านเราก็สามารถสร้างระบบของมันให้เป็นระบบ Client/Server หรือ Peer-to-Peer processing ได้แล้ว ในระบบเมนเฟรมของ IBM เองก็เริ่มต้นความสามารถทางด้าน Peer-to-Peer networking ขึ้นมาแล้วใน NCP และ VTAM เวอร์ชันปัจจุบัน

แต่ระบบแบบ Client/Server และ Peer-to-Peer processing ยังจะต้องได้รับการพัฒนาให้ดียิ่งขึ้นไปอีก เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพที่ดีมากโดยที่ใช้รีซอร์สต่าง ๆ ให้น้อยลง ผู้เขียนมีความเห็นว่าภายในทศวรรษที่ 90 นี้ คุณจะได้เห็นระบบมากมายเปลี่ยนตัวเองผ่านเข้าไปเป็นแบบ Client/Server และไปสิ้นสุดที่จุดหมายเดียวกันคือการทำงานแบบ Peer-to-Peer processing

ในระบบเครือข่ายเองก็เร่งที่จะพัฒนามาเป็นระบบ Peer-to-Peer กันมากขึ้น ซึ่งคุณจะเห็นได้จากการที่มีระบบปฏิบัติการระบบแบบใหม่ ๆ ที่หันมาใช้การทำงานแบบ Peer-to-Peer processing แอพพลิเคชันที่วิ่งอยู่ในเซอร์ฟเวอร์ก็เป็นแบบ Client/Server กันเป็นส่วนใหญ่ ดังนั้นถ้าคุณตัดสินใจที่จะพัฒนาแอพพลิเคชันสักตัว คุณก็คงจะเห็นความสำคัญของการทำงานแบบ Client/Server และ Peer-to-Peer processing และเริ่มพัฒนาให้ออกไปในแนวทางนี้เพื่อทำให้ระบบเครือข่ายและแอพพลิเคชันของคุณทำงานได้อย่างมีประสิทธิภาพมากขึ้น

สำหรับหน่วยงานหรือสถานศึกษาที่มีระบบเมนเฟรม ระบบมินิ และระบบเครือข่ายใช้อยู่ร่วมกัน รูปแบบของการก้าวไปในการพัฒนาการทำงานของแอพพลิเคชันก็จะเป็นการผสมผสานกันระหว่างระบบทั้งหมดที่จะทำงานภายใต้พื้นฐานอันเดียวกัน

ผู้ใช้สามารถร้องขอข้อมูลต่าง ๆ ไม่ว่าจะอยู่ในระบบเมนเฟรม มินิ หรือในเซอร์ฟเวอร์ในระบบเครือข่ายได้อย่างง่ายดายโดยไม่ต้องสนใจเลยว่าข้อมูลที่ตนต้องการอยู่ในระบบใด ซึ่งเป็นการเริ่มต้นที่จะรวมเอาการทำงานทั้งหมดเข้าด้วยกัน จากค่าบำรุงรักษาและราคาซอฟต์แวร์ของระบบเมนเฟรมและมินิที่มีราคาแพงมาก ผู้ควบคุมระบบก็จะค่อย ๆ ดึงเอาฐานข้อมูลใหญ่ ๆ ที่อยู่ในระบบเมนเฟรมหรือมินิลงมาอยู่ในระบบเครือข่ายหรือตัวเซอร์ฟเวอร์เพื่อกระจายการทำงานออกและลดจำนวนระบบเมนเฟรมหรือมินิลงเพื่อประหยัดค่าใช้จ่ายให้น้อยลง การขยายตัวของระบบแบบ Client/Server ก็จะทำได้อย่างง่ายดายขึ้นและเสียค่าใช้จ่ายน้อยกว่าด้วย